Sampling Algebraic Sets in Local Intrinsic Coordinates*

Yun Guan† Jan Verschelde‡

14 December 2009

Abstract

Numerical data structures for positive dimensional solution sets of polynomial systems are sets of generic points cut out by random planes of complimentary dimension. We may represent the linear spaces defined by those planes either by explicit linear equations or in parametric form. These descriptions are respectively called extrinsic and intrinsic representations. While intrinsic representations lower the cost of the linear algebra operations, we observe worse condition numbers. In this paper we describe the local adaptation of intrinsic coordinates to improve the numerical conditioning of sampling algebraic sets. Local intrinsic coordinates also lead to a better stepsize control. We illustrate our results with Maple experiments and computations with PHCpack on some benchmark polynomial systems.

2000 Mathematics Subject Classification. Primary 65H10. Secondary 14Q99, 68W30.
Key words and phrases. algebraic sets, condition numbers, generic points, local intrinsic coordinates, numerical algebraic geometry, path tracking, polynomial systems, sampling.

1 Motivation, Definitions, and Problem Statement

A polynomial system $f(\mathbf{x}) = \mathbf{0}$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, defines an algebraic set $f^{-1}(\mathbf{0}) \subset \mathbb{C}^n$. The polynomials of f belong to $\mathbb{C}[\mathbf{x}]$. We assume (for simplicity of exposition throughout the paper):

1. $f^{-1}(\mathbf{0})$ is pure dimensional, k is its codimension, so $\dim(f^{-1}(\mathbf{0})) = n - k$;
2. $f(\mathbf{x}) = \mathbf{0}$ is a complete intersection, and in particular: $f = (f_1, f_2, \ldots, f_k)$;
3. $f^{-1}(\mathbf{0})$ is reduced, i.e.: of multiplicity one.

To remove the third assumption, a deflation operator [21] (see also [10]) as proposed in [32, §13.3.2] should be applied. The first two assumptions are made for notational convenience.

The numerical treatment of positive dimensional algebraic sets was first proposed in [31] and elaborated in a series of papers by the authors of [32] and the second author, see also [30]

*This material is based upon work supported by the National Science Foundation under Grant No. 0713018.

†Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan (M/C 249), Chicago, IL 60607-7045, USA. email: guan@math.uic.edu URL: http://www.math.uic.edu/~guan

‡Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan (M/C 249), Chicago, IL 60607-7045, USA. email: jan@math.uic.edu URL: http://www.math.uic.edu/~jan

for another introduction. The algorithms in numerical algebraic geometry are implemented in PHCpack [33] and Bertini [4] (see [6] and [26]) and can be executed via MATLAB (or Octave) [15], Maple [20], and Macaulay 2 [19].

One of our benchmark examples is a family of systems, defined by all adjacent minors of a general 2-by-3 matrix ([12], [16]):

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \quad f(\mathbf{x}) = \begin{cases} x_{11}x_{22} - x_{21}x_{12} = 0 \\ x_{12}x_{23} - x_{22}x_{13} = 0. \end{cases} \tag{1}$$

For this example, we have $n = 6$, $k = 2$, and we have a complete intersection: $\dim(f^{-1}(\mathbf{0})) = n - k = 4$. To compute $\deg(f^{-1}(\mathbf{0}))$, we add $n - k$ general linear equations $L(\mathbf{x}) = \mathbf{0}$ to $f(\mathbf{x}) = \mathbf{0}$ and solve $\{f(\mathbf{x}) = \mathbf{0}, L(\mathbf{x}) = \mathbf{0}\}$. Generic points on the solution set defined by the system for all adjacent minors of a general 2-by-3 matrix satisfy (for random coefficients $c_{ij} \in \mathbb{C}$):

$$\begin{cases} x_{11}x_{22} - x_{21}x_{12} = 0 \\ x_{12}x_{23} - x_{22}x_{13} = 0 \\ c_{10} + c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{14}x_{21} + c_{15}x_{22} + c_{16}x_{23} = 0 \\ c_{20} + c_{21}x_{11} + c_{22}x_{12} + c_{23}x_{13} + c_{24}x_{21} + c_{25}x_{22} + c_{26}x_{23} = 0 \\ c_{30} + c_{31}x_{11} + c_{32}x_{12} + c_{33}x_{13} + c_{34}x_{21} + c_{35}x_{22} + c_{36}x_{23} = 0 \\ c_{40} + c_{41}x_{11} + c_{42}x_{12} + c_{43}x_{13} + c_{44}x_{21} + c_{45}x_{22} + c_{46}x_{23} = 0. \end{cases} \tag{2}$$

Except for an algebraic set in the coefficient space c_{ij} for L, the system above has four solutions, we have four generic points for all adjacent minors of a general 2-by-3 matrix, so $\deg(f^{-1}(\mathbf{0})) = 4$.

To save work, reducing the number of variables from 6 to 2, we choose a different representation for the linear space defined by the equations $L(\mathbf{x}) = \mathbf{0}$, representing the 2-plane $L^{-1}(\mathbf{0})$ in $\mathbb{C}^6$ as

$$\begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} + \xi_1 \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \\ v_{15} \\ v_{16} \end{bmatrix} + \xi_2 \begin{bmatrix} v_{21} \\ v_{22} \\ v_{23} \\ v_{24} \\ v_{25} \\ v_{26} \end{bmatrix} \tag{3}$$

spanned by an offset point $\mathbf{b} \in \mathbb{C}^6$ and an orthonormal basis $\{\mathbf{v}_1, \mathbf{v}_2\}$. The tuple (ξ_1, ξ_2) defines *intrinsic coordinates* for the generic points, introduced in [29] to speedup the algorithms of [28].

The reduction from six to two variables reduces the cost of solving linear systems by a factor of nine. This reduction improves the efficiency of Newton's method when computing sample points on the algebraic set, one of the basic operations in numerical algebraic geometry [32].

For any $f^{-1}(\mathbf{0}) \in \mathbb{C}^n$ with $\dim(f^{-1}(\mathbf{0})) = n - k$, we use a general k-plane L to compute generic points. This general k-plane L may be defined in two equivalent ways:

1. $L(\mathbf{x}) = 0$ is a system of $n - k$ general linear equations in $\mathbf{x}$,
2. $\mathbf{b} \in \mathbb{C}^n$ is an offset point, and $V = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_k] \in \mathbb{C}^{n \times k}$, with $V^*V = I_k$, i.e.: V is an orthonormal[1] basis of vectors.

[1]Although it suffices to require that the columns of the matrix V are linearly independent, the orthonormality condition $V^*V = I_k$ (using complex conjugated inner products and I_k is the k-by-k identity matrix) is beneficial.

If linear equations define L, solving $\{f(\mathbf{x}) = \mathbf{0}, L(\mathbf{x}) = \mathbf{0}\}$ gives generic points in their usual form that we call an extrinsic coordinate representation. Using $(\mathbf{b}, V)$ for L gives intrinsic coordinates $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_k)$ for generic points $\mathbf{x}$:

$$\mathbf{x} = \mathbf{b} + \xi_1 \mathbf{v}_1 + \xi_2 \mathbf{v}_2 + \cdots + \xi_k \mathbf{v}_k = \mathbf{b} + V\boldsymbol{\xi}. \tag{4}$$

With intrinsic coordinates for generic points, the original variables $\mathbf{x}$ become place holders when solving $f(\mathbf{x} = \mathbf{b} + V\boldsymbol{\xi}) = \mathbf{0}$. In Figure 1, we outline the two ways to compute generic points.

$$\begin{array}{ccc} L & \xrightarrow{\quad K_E \quad} & \mathbf{x} \\ \downarrow & & \uparrow \\ (\mathbf{b}, V) & \xrightarrow{\quad K_I \quad} & \boldsymbol{\xi} \end{array}$$

Figure 1: A commutative diagram for extrinsic $\mathbf{x}$ and intrinsic $\boldsymbol{\xi}$ coordinates of generic points. The vertical arrows require linear algebra while the horizontal arrows involve the solution of polynomial systems. Their sensitivities are determined by condition numbers K_E and K_I.

In shorthand notation, the general k-plane L is represented as $(\mathbf{b}, V)$ and we use intrinsic coordinates $\boldsymbol{\xi} \in \mathbb{C}^k$ to denote the generic points. When sampling points, the moving L from $(\mathbf{b}, V)$ to $(\mathbf{c}, W)$, is done via the obvious homotopy:

$$f\left(\begin{array}{ccccc} \mathbf{x} = & (1-t)\mathbf{b} + t\mathbf{c} & + & ((1-t)V + tW) & \boldsymbol{\xi} \\ & \text{moving offset point} & & \text{moving basis vectors} & \end{array} \right) = \mathbf{0}. \tag{5}$$

As t moves from 0 to 1, the solution paths $\boldsymbol{\xi}(t)$ are tracked with predictor-corrector methods and give new generic points on $f^{-1}(\mathbf{0})$. For introductions to path following and continuation methods we refer to [1] and [24], see also [22].

While the diagram in Figure 1 commutes for exact operations, using floating-point arithmetic forces us to take into account condition numbers. These condition numbers bound the growth of the relative errors on the solutions as a consequence of relative errors on the input data. As we keep f fixed during the computation we only consider relative errors on the representations of the k-plane L. Formally, we introduce condition numbers K_E and K_I on the extrinsic and intrinsic coordinate representations respectively as

$$\frac{||\Delta \mathbf{x}||}{||\mathbf{x}||} \leq K_E \frac{||\Delta L||}{||L||} \quad \text{and} \quad \frac{||\Delta \boldsymbol{\xi}||}{||\boldsymbol{\xi}||} \leq K_I \frac{||\Delta(\mathbf{b}, V)||}{||(\mathbf{b}, V)||}. \tag{6}$$

Going to intrinsic coordinates, we observe a worsening of the numerical conditioning: $K_I \gg K_E$. Note that the original problem is well conditioned, in other words, K_E is expected to remain small, because random choices for L avoids places where the Jacobian matrix of f drops rank.

To get a first intuition why $K_I \gg K_E$, consider the binomial expansion of a monomial $x_1^{a_1} x_2^{a_2}$ of f. If we evaluate $x_1^{a_1} x_2^{a_2}$ at $x_1 = b_1 + \xi_1 v_1$ and $x_2 = b_2 + \xi_2 v_2$, we compute

$$(b_1 + \xi_1 v_1)^{a_1} (b_2 + \xi_2 v_2)^{a_2} = \left(\sum_{i=0}^{a_1} \binom{a_1}{i} b_1^i (\xi_1 v_1)^{a_1 - i} \right) \left(\sum_{j=0}^{a_2} \binom{a_2}{j} b_2^j (\xi_2 v_2)^{a_2 - j} \right) \tag{7}$$

and we see that any sparse structure of f will be destroyed. Moreover, the binomial coefficients in (7) inflate the variation among the coefficients in f.

In general, we may write $f(\mathbf{b}+V(\boldsymbol{\xi}+\Delta\boldsymbol{\xi}))=f(\mathbf{b}+V\boldsymbol{\xi})+\Delta f$. The trouble is that, even for small $||\Delta\boldsymbol{\xi}||$ we may experience very large $||\Delta f||$.

While using multiprecision arithmetic during path tracking [5] may avoid these numerical instabilities, using multiprecision numbers significantly slows down the computations and when the coefficients are known with limited accuracy, applying multiprecision arithmetic may give misleading answers. Better stepsize control strategies [7] will also be effective for our problems, but like in dealing with the high powers of the continuation parameter of polyhedral homotopies [18], our approach in this paper is specific to the type of homotopies. To deal with the numerical instabilities of using intrinsic coordinates, we propose the use of *local* intrinsic coordinates. We define local coordinates in the next section. In section 3, we present an algorithm to track a solution path using intrinsic coordinates, along with an a priori stepsize control evaluation strategy. Computational results are discussed in the section 4.

Acknowledgement. We thank Professor Hiroshi Murakami for his remarks made after the presentation of the first author at the session of Symbolic and Numeric Computation at ACA 2009. His remarks led us to local intrinsic coordinates.

2 Local Intrinsic Coordinates

In this section we define local intrinsic coordinate representations of generic points and address the improved numerical conditioning.

What if we could keep $||\boldsymbol{\xi}||$ small? Writing Greek symbols badly, the ξ looks close enough to an epsilon, and then reconsidering the binomial expansions in (7):

$$\begin{aligned}(b_1+\xi_1v_1)^{a_1}\,(b_2+\xi_2v_2)^{a_2} &= \left(b_1^{a_1}+a_1b_1^{a_1-1}\xi_1v_1+O(\xi_1^2)\right)\left(b_2^{a_2}+a_2b_2^{a_2-1}\xi_2v_2+O(\xi_2^2)\right) \quad (8)\\ &= b_1^{a_1}b_2^{a_2}+a_1b_1^{a_1-1}b_2^{a_2}\xi_1v_1+a_2b_1^{a_1}b_2^{a_2-1}\xi_2v_2+O(\xi_1^2,\xi_1\xi_2,\xi_2^2). \quad (9)\end{aligned}$$

If we assume that ξ is infinitesimally small, then we ignore the second order terms $O(\xi_1^2,\xi_1\xi_2,\xi_2^2)$.

For general polynomials f, writing $f(\mathbf{b}+V\boldsymbol{\xi})=f(\mathbf{b})+\Delta f$, the omission of the higher order terms leads to: $||\Delta f||$ is $O(||V\boldsymbol{\xi}||)$. Because we may select for the orthonormal basis V a nice numerical representation, we have that $O(||V\boldsymbol{\xi}||)$ is $O(||\boldsymbol{\xi}||)$ and therefore: $||\Delta f||$ is $||O(\boldsymbol{\xi})||$.

To keep $||\boldsymbol{\xi}||$ small, we now propose to use the extrinsic coordinates of the generic point as the offset point for a k-plane. In particular, for $d=\deg(f^{-1}(\mathbf{0}))$ and d generic points $\{\mathbf{z}_1,\mathbf{z}_1,\ldots,\mathbf{z}_d\}$ on $f^{-1}(\mathbf{0})$, consider:

$$\mathbf{x}=\mathbf{z}_\ell+V\boldsymbol{\xi},\quad \ell=1,2,\ldots,d. \tag{10}$$

Because $L(\mathbf{z}_\ell)=\mathbf{0}$ for all generic points, all $\mathbf{z}_\ell+V\boldsymbol{\xi}$ represent the same k-plane L. Given an orthonormal basis V for a k-plane and a set $\{\mathbf{z}_1,\mathbf{z}_1,\ldots,\mathbf{z}_d\}$ of d generic points on $f^{-1}(\mathbf{0})$, the *local intrinsic coordinates* to represent $f^{-1}(\mathbf{0})$ are defined by the tuple $(\{\mathbf{z}_1,\mathbf{z}_1,\ldots,\mathbf{z}_d\},V)$.

Obviously, the transition from *global* intrinsic coordinates $\mathbf{x}=\mathbf{b}+V\boldsymbol{\xi}$ to *local* intrinsic coordinates is performed by a mere evaluation of $\mathbf{b}+V\boldsymbol{\xi}$. The close relation between local intrinsic coordinates and extrinsic coordinates will yield improved condition numbers.

To define the condition number K_E of a zero $\mathbf{z}$ of $F := \{f(\mathbf{x}) = \mathbf{0}, L(\mathbf{x}) = \mathbf{0}\}$, we consider the application of Newton's method:

$$\underbrace{F'(\mathbf{z})}_{=A} \Delta\mathbf{z} = -F(\mathbf{z}), \quad K_E := \kappa(A), \tag{11}$$

where F' is the matrix of all partial derivatives of F and $\kappa(A)$ is the condition number of the Jacobian matrix A of F at $\mathbf{z}$. Because we assume that $f(\mathbf{x}) = \mathbf{0}$ is a complete intersection, $A\Delta\mathbf{z} = -F(\mathbf{z})$ is a well defined n-by-n linear system. Strictly speaking, as we keep f fixed and vary only the linear equations $L(\mathbf{x}) = \mathbf{0}$, we will have $K_E \leq \kappa(A)$, but because generic points are always well conditioned this distinction is very minor.

In local intrinsic coordinates we replace $\mathbf{x}$ by $\mathbf{z} + V\boldsymbol{\xi}$ and the application of Newton's method leads to

$$\underbrace{f'(\mathbf{z} + V\boldsymbol{\xi})}_{=B} \Delta\boldsymbol{\xi} = -f(\mathbf{z} + V\boldsymbol{\xi}), \quad K_{LI} := \kappa(B), \tag{12}$$

where f' is the matrix of all partial derivatives of f and $\kappa(B)$ is the condition number of the Jacobian matrix B of f at $\boldsymbol{\xi}$. Because we assume that $f(\mathbf{x}) = \mathbf{0}$ is a complete intersection, $B\Delta\boldsymbol{\xi} = -f(\boldsymbol{\xi})$ is a well defined k-by-k linear system. We define $\kappa(B)$ as K_{LI}, the condition number of $\mathbf{z}$ represented in local intrinsic coordinates.

Observe the similarity of (11) with (12) as we write (11) more explicitly as

$$\begin{bmatrix} f'(\mathbf{z}) \\ L'(\mathbf{z}) \end{bmatrix} \Delta\mathbf{z} = -\begin{bmatrix} f(\mathbf{z}) \\ L(\mathbf{z}) \end{bmatrix}, \tag{13}$$

where L' contains all partial derivatives of L.

Proposition 2.1 *With K_E and K_{LI} as defined in* (11) *and* (12) *respectively: $K_{LI} \approx K_E$.*

Proof. Because in local intrinsic coordinates: $\boldsymbol{\xi} = \mathbf{0}$, it does no longer make sense to consider relative errors. Moreover, without loss of generality we may always choose coefficients of the planes so that $||L|| = 1$ and $||V|| = 1$. Using homogeneous coordinates for f, we assume we work in an appropriate affine chart so that also $||\mathbf{z}|| = 1$. Then the meaning for the condition numbers K_E and K_{LI} are in the inequalities

$$||\Delta\mathbf{z}|| \leq K_E||\Delta L|| \quad \text{and} \quad ||\Delta\boldsymbol{\xi}|| \leq K_{LI}||\Delta(\mathbf{b}, V)||, \tag{14}$$

where $\mathbf{b}$ is an offset point and V are directions in a parametric representation of L.

Using the commutative diagram of Figure 1, we relate $\Delta\mathbf{z}$ and $\Delta\boldsymbol{\xi}$:

$$\mathbf{z} + V(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}) = \mathbf{z} + \Delta\mathbf{z} \quad \Rightarrow \quad \Delta\mathbf{z} = V\Delta\boldsymbol{\xi} \quad \text{as } \boldsymbol{\xi} = \mathbf{0}. \tag{15}$$

Looking at norms:

$$||\Delta\mathbf{z}|| = ||V\Delta\boldsymbol{\xi}|| = ||\Delta\boldsymbol{\xi}|| \quad \text{as } ||V|| = 1. \tag{16}$$

Because $V^*V = I_k$, all eigenvalues of V lie on the complex unit circle and multiplication with V is norm preserving.

In local intrinsic coordinates, for $\boldsymbol{\xi} = \mathbf{0}$, changes ΔV in the orientation of L do not influence $\boldsymbol{\xi}$. So we have $\Delta(\mathbf{b}, V) = \Delta\mathbf{b}$ and in case we may consider $||\Delta L|| \approx ||\Delta(\mathbf{b}, V)||$. Thus in (14) we may interchange K_E with K_{LI}, so $K_{LI} \approx K_E$. □

3 A Rescaling Algorithm

In this section we consider the sampling of algebraic sets using local intrinsic coordinates. We define a rescaling algorithm and address its numerical stability. In addition, using local intrinsic coordinates leads to a better stepsize control.

Generic points $\{\mathbf{z}_1, \mathbf{z}_1, \ldots, \mathbf{z}_d\}$ are offset points for a k-plane L with directions in the orthonormal matrix V. In local intrinsic coordinates, moving from $(\mathbf{z}_\ell, V)$ to $(\mathbf{b}, W)$, as t goes from 0 to 1, the deformations are defined by

$$f\left(\mathbf{x} = (1-t)\mathbf{z}_\ell + t\mathbf{b} + W\boldsymbol{\xi}\right) = \mathbf{0}. \tag{17}$$

In contrast to the obvious homotopy in (5), we see that only the offset point moves. We immediately switched from the current directions in V to the new orthonormal basis W because $\boldsymbol{\xi} = \mathbf{0}$ in local intrinsic coordinates. But this is only a first indication of the potential of working with local intrinsic coordinates, we can do better than (17).

Instead of using (17) and moving to $\mathbf{b}$, we point out that any point in the k-plane L can serve as an offset point. Therefore, we should choose the best offset point, i.e.: the point closest to the current generic point. To compute the closest point, let $\mathbf{c}$ be the orthogonal projection of $\mathbf{z}_\ell$ onto the k-plane L. For some step size h, we then consider:

$$f\left(\mathbf{x} = \mathbf{z}_\ell + h(\mathbf{c} - \mathbf{z}_\ell) + W\boldsymbol{\xi}\right) = \mathbf{0} \tag{18}$$

and apply Newton's method to find the correction $\Delta\boldsymbol{\xi}$, as illustrated in Figure 2.

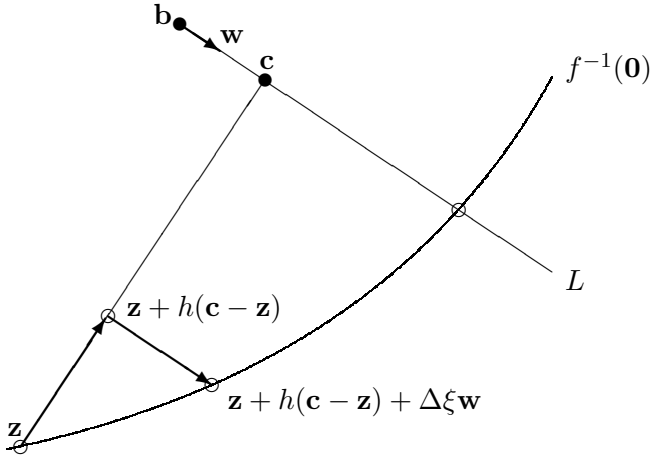

Figure 2: Schematic of one predictor-corrector step of the new sampling algorithm, moving from the point $\mathbf{z}$ to the point where L meets $f^{-1}(\mathbf{0})$. The line L is defined by $\mathbf{b} + \xi\mathbf{w}$. Using step size h, the prediction $h(\mathbf{c} - \mathbf{z})$ added to $\mathbf{z}$ occurs in a direction orthogonal to L while the correction $\Delta\xi\mathbf{w}$ is parallel to L.

After each step, we add the correction term ($\Delta\xi\mathbf{w}$ in Figure 2) to the offset point, rescaling the intrinsic coordinates to local intrinsic coordinates at the end of the correction stage. Pseudocode

for one predictor-corrector step is given in Algorithm 3.1, going from one generic point $\mathbf{z} \in f^{-1}(\mathbf{0}) \cap K$, where K is the current k-plane, towards L the target k-plane.

Algorithm 3.1 (one predictor-corrector step in local intrinsic coordinates)

Input:	$f = (f_1, f_2, \ldots, f_k)$, $f_i(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$, $i = 1, 2, \ldots, k$;	$\dim(f^{-1}(\mathbf{0})) = n - k$
	$\mathbf{b} \in \mathbb{C}^n$;	*offset point of k-plane L*
	$W = [\mathbf{w}_1 \ \mathbf{w}_2 \ \cdots \ \mathbf{w}_k] \in \mathbb{C}^{n \times k}$, $W^*W = I_k$;	*orthonormal basis for L*
	$\mathbf{z} \in \mathbb{C}^n$: $f(\mathbf{z}) = \mathbf{0}$, $K(\mathbf{z}) = \mathbf{0}$;	*generic point on k-plane K*
	$h > 0$;	*step size*
	$\epsilon > 0$.	*accuracy requirement*
Output:	$\widehat{\mathbf{z}}$, $f(\widehat{\mathbf{z}}) = \mathbf{0}$, $L(\widehat{\mathbf{z}}) = \mathbf{0}$: $\|\|\widehat{\mathbf{z}} - \mathbf{b}\|\| < \|\|\mathbf{z} - \mathbf{b}\|\|$.	*generic point closer to L*
1.	$\mathbf{v} := \mathbf{z} - \mathbf{b}$;	*go towards offset point*
2.	$\mathbf{v} := \mathbf{v} - \sum_{i=1}^{k} (\overline{\mathbf{w}_i}^T \mathbf{v}) \mathbf{w}_i$;	*move perpendicular to L*
3.	$\mathbf{v} := \frac{\mathbf{v}}{\|\|\mathbf{v}\|\|}$;	*normalize so $\|\|\mathbf{v}\|\| = 1$*
4.	$\widetilde{\mathbf{z}} := \mathbf{z} + h\ \mathbf{v}$;	*prediction for new generic point*
5.	$\widehat{\mathbf{z}} := \widetilde{\mathbf{z}}$; $\boldsymbol{\xi} := \mathbf{0}$;	*initialize for Newton corrector*
6.	while $\|\|f(\widehat{\mathbf{z}} + W\boldsymbol{\xi})\|\| > \epsilon$ do	*as long as not accurate enough*
6.1	$\Delta\boldsymbol{\xi} := f(\widehat{\mathbf{z}} + W\boldsymbol{\xi}) / f'(\widehat{\mathbf{z}} + W\boldsymbol{\xi})$;	*solve a linear system for $\Delta\boldsymbol{\xi}$*
6.2	$\boldsymbol{\xi} := \boldsymbol{\xi} + \Delta\boldsymbol{\xi}$;	*update correction*
7.	$\widehat{\mathbf{z}} := \widehat{\mathbf{z}} + W\boldsymbol{\xi}$.	*rescale to local coordinates*

The orthonormality condition $W^*W = I_k$ is important for instruction 2 in the Algorithm 3.1 because we can compute the projection just via inner products. The number of arithmetical operations needed to carry out instruction 2 in Algorithm 3.1 is $O(kn)$. Without the condition $W^*W = I_k$, this cost (e.g. via Gram-Schmidt orthogonalization) would be at least $O(kn^2)$.

For the numerical stability of Algorithm 3.1, we first discuss the relationship between the step size h and the accuracy requirement ϵ. If on the one hand h is too small, then the condition $||f(\widehat{\mathbf{z}} + W\boldsymbol{\xi})|| > \epsilon$ in the while-do instruction 6 of Algorithm 3.1 is directly satisfied. On the other hand, if h is too large, satisfying the accuracy requirement of instruction 6 may require too many iterations, or Newton's method may not converge at all. We point out that the cost of instruction 6.1 is $O(k^3)$ and if f is sufficiently sparse (if evaluation and differentiation go fast), then the cost of execution of Newton's method dominates the cost of Algorithm 3.1.

In general path tracking algorithms, the step size h is determined via a feedback mechanism. If Newton's method does not converge fast enough, then the step size is reduced. If Newton's method needs only two steps or less, then the step size might be enlarged. See [7] for stepsize control strategies. The problem with this feedback mechanism is that it comes at the great expense of the most costly portion of the predictor-corrector method, i.e.: each reduction of h comes at the expense of a failed and thus wasted Newton step. With local intrinsic coordinates, we can predict the fitness of the step size with a simple evaluation. For some step size h and direction $\mathbf{v}$, we evaluate and estimate the residual as

$$||f(\mathbf{x} = \mathbf{z}_\ell + h\mathbf{v})|| \text{ is } ||f(\mathbf{z}_\ell) + O(h)|| \text{ is } O(h). \tag{19}$$

For example, if $h = 10^{-2}$ and we see that the residual is $O(10^{-2})$, then it is fair to expect that after one iteration of Newton's method, the residual becomes $O(10^{-4})$, and then $O(10^{-8})$ after the second iteration.

In Algorithm 3.2 we define how to cut back on the step size just by evaluation, *before* the start of the Newton correction.

Algorithm 3.2 (a priori stepsize control by evaluation)

Input: $f = (f_1, f_2, \ldots, f_k)$, $f_i(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$ $i = 1, 2, \ldots, k$;	$\dim(f^{-1}(\mathbf{0})) = n - k$
$\mathbf{z} \in \mathbb{C}^n$: $f(\mathbf{z}) = \mathbf{0}$, $K(\mathbf{z}) = \mathbf{0}$;	*generic point on k-plane K*
$\mathbf{v} \in \mathbb{C}^n$, $\|\|\mathbf{v}\|\| = 1$;	*direction vector*
$h > 0$;	*current step size*
$\delta > 0$.	*threshold to reduce h*
$1 > \rho > 0$.	*reduction factor for h*
Output: $h > 0$.	*updated step size*
1. $y := \|\|f(\mathbf{z} + h\mathbf{v})\|\|$;	*evaluate the predicted point*
2. if $y/h > \delta$ then $h := \rho h$.	*reduce the step size*

The reduction of the step size in instruction 2 of Algorithm 3.2 could be followed by another evaluation of f to see if y is reduced sufficiently or has become even too small.

Algorithm 3.2 is called after instruction 3 of Algorithm 3.1.

By application of Algorithm 3.2, occurrences of a diverging Newton's method can be greatly reduced because the size of the residual $||f(\mathbf{x} = \mathbf{z} + W\boldsymbol{\xi})||$ is $O(h)$.

We conclude with a quick cost estimate for the total number of Newton steps along one path. In sampling for generic points, we typically choose the new random coefficients for the k-plane as complex numbers on the unit circle, so the distance between two k-planes (and in particular their offset points) is $O(1)$. For h: $0 < h < 1$, we can see that the total number of Newton iterations along a solution path is proportional to $1/h$. For example if $h = 0.01$ and we need about 2 or 3 Newton iterations per step, then the total number of Newton iterations along a solution path will vary between 200 and 300.

The homotopy continuation methods of this paper are different from the so-called linear homotopies for which an experimental study to certify path tracking recently appeared in [8]. A potential future research direction could be to expand the quick cost estimate of the previous paragraph into a formal complexity study, along the lines of [9] and [25].

4 Computational Results

Local intrinsic coordinates are available in version 2.3.53 of PHCpack [33]. We first describe numerical experiments done with Maple to compare condition numbers of generic points on a hypersurface of polynomials of increasing degrees. Then we report preliminary results on small benchmark problems with the sampling routines in PHCpack. All computations were done on one core of a Mac OS X 3.2 Ghz Intel Xeon.

4.1 Condition Number Estimates

In this section we illustrate the worsening of the conditioning of using global intrinsic coordinates on one sparse polynomial. We give data on sampling with zero and nonzero offset vectors and relate this experiment to using local intrinsic coordinates.

To estimate the condition numbers we use `LinearAlgebra[EigenConditionNumbers]` of Maple 12, with `UseHardwareFloats` set to `true`, see [23, Chapter 4]. The corresponding documentation pages of Maple 12 refer to [2]. For an introduction to the perturbation theory of eigenvalues, see e.g.: [11, §4.3].

We consider one sparse polynomial f in $n = 10$ variables, of increasing degrees d, starting with t terms. In addition, we add all the linear terms $c_i x_i$, $i = 1, 2, \ldots, n$, to avoid ending up with the origin as a multiple root. The coefficients are taken on the complex unit circle. The particular Maple commands used to generate an f are

```
[> n := 10: d := 10: t := 5:
[> c := () -> exp(I*stats[random,uniform[0,2*Pi]](1)):
[> X := [seq(x[i],i=1..n)]:
[> f := X[1]^d + randpoly(X,coeffs=c,degree=d-1,terms=5) + sum(c()*x[i],i=1..n);
```

The first term of `f` ensures that we have a monic polynomial after substitution $f(\mathbf{v}\xi)$, for $v_1 = 1$. That f is monic is convenient for the connection with the companion matrix when we look at the condition numbers of the corresponding eigenvalue problem.

To introduce the idea of using different coordinate systems, we respectively use

$$\mathbf{x} = \mathbf{b} + \mathbf{v}\xi \quad \text{and} \quad \mathbf{x} = \mathbf{v}\xi, \quad \mathbf{b}, \mathbf{v} \in \mathbb{C}^n, \tag{20}$$

where all coefficients in the vectors are also taken on the complex unit circle. With $f(\mathbf{v}\xi) = 0$ we obtain still a sparse polynomial with all coefficients on the complex unit circle, which is not the case with $f(\mathbf{b} + \mathbf{v}\xi) = 0$. The offset vector of $\mathbf{b} + \mathbf{v}\xi$ is responsible for the variation in the coefficients and the fluctuation of the condition numbers we observe in our numerical experiments, summarized in Table 1.

degrees of f	$f(\mathbf{b}+\mathbf{v}\xi)=0$ largest	$f(\mathbf{b}+\mathbf{v}\xi)=0$ smallest	$f(\mathbf{v}\xi)=0$ largest	$f(\mathbf{v}\xi)=0$ smallest	ratios of smallest	ratios of largest
10	5.91e-01	9.02e-02	8.81e-01	4.01e-01	6.55e+00	2.20e+00
20	2.77e-01	1.76e-03	8.91e-01	3.31e-01	1.57e+02	2.70e+00
30	2.75e-01	6.16e-05	9.49e-01	7.25e-02	4.47e+03	1.31e+01
40	4.53e-01	7.14e-06	9.69e-01	1.87e-01	6.34e+04	5.17e+00

Table 1: Estimates for the inverse condition numbers of eigenvalues of the companion matrices of $f(\mathbf{b}+\mathbf{v}\xi) = 0$ and $f(\mathbf{v}\xi) = 0$. For degrees $d = 10$, 20, 30, and 40, we list the largest and smallest inverse condition numbers. For $f(\mathbf{v}\xi) = 0$, we see the range between smallest and largest not widen that much, whereas for $f(\mathbf{b}+\mathbf{v}\xi) = 0$, the conditioning steadily worsens.

As we see from Table 1, all roots of $f(\mathbf{v}\xi) = 0$ are well conditioned. To compare the conditioning of local intrinsic coordinates, we take the first root z_1 of $f(\mathbf{v}\xi) = 0$ and consider the

companion matrix A of $f(z_1 + \mathbf{v}\xi) = 0$. For increasing degrees, the condition number for the zero $\xi = 0$ corresponding to z_1 is always reported as `1.00e+00`. The smallest inverse condition numbers of the eigenvalues of A for degrees $d = 10$, 20, and 30 are respectively `8.42e-05`, `1.08e-12`, and `3.69e-14`. This implies that for $d = 30$ we have lost all accuracy as our working precision are the standard hardware floats.

In this simple Maple experiment we illustrate that, although sampling a hypersurface is reduced to solving univariate polynomial equations, for hypersurfaces defined by polynomials of high degrees we cannot use the same representation of a general line to define generic points. If we adapt the offset point and switch to local intrinsic coordinates, then the generic points are well conditioned.

4.2 Sampling Benchmark Systems

The input to the sampling problem is one set of generic points on $f^{-1}(\mathbf{0}) \cap L$ and a new k-plane K. On output is a new set of generic points on $f^{-1}(\mathbf{0}) \cap K$.

The polynomial systems we selected occur in the literature. We briefly summarize the main characteristics of these systems:

1. All adjacent minors of a general 2-by-n matrix, $n = 3, 4, \ldots$. This is a family of nice quadratic equations arising in algebraic statistics [12].

2. The cyclic n-roots systems are well known academic benchmarks. If n has a quadratic divisor, then the system has a positive dimensional solution set [3]. In our experiments we use the cyclic 8-roots system, which has a one dimensional solution set of degree 144.

3. Griffis-Duffy platforms [13] are architecturally singular mechanisms [17], their motion correspond to curves of degree 40 in 8-space [27].

For the purposes of this paper, the computation of the first set of generic points is considered as given, typically in extrinsic coordinate representation.

Except for the adjacent minors, the systems are not complete intersections. For $m > k$, to make an m-by-k system f square, we generate a random k-by-m matrix C and work with $C \times f$.

To test the improvement from using local intrinsic coordinates, we sample new generic points from the solution sets. Our computational experimental setup consists of three stages:
(1) Given one set of generic points, we generate another random k-plane L.
(2) We then move the given set of generic points to lie on L.
(3) At the end we check results for accuracy, count #predictor-corrector steps, record elapsed cpu times.

Note that the recorded cpu times are only meant to give an indication on the relative practical difficulties of these problems. More relevant are the number of iterations performed by Newton's method along the paths.

In Table 2 we summarize the results. Even as the systems we selected as benchmark examples are not challenging, we observe a clear benefit of using local intrinsic coordinates, even for the systems defined by quadratic equations. The benefit is perhaps most significant for the cyclic 8-roots problem where the degree of the ith polynomial equals i.

polynomial system	n	$n-k$	d	#iterations	timings
Griffis-Duffy platform	8	1	40	207/164	550/535 μsec
cyclic 8-roots system	8	1	144	319/174	5.3/3.2 sec
all adjacent minors	22	12	1,024	285/219	44.6/40.3 sec

Table 2: Preliminary experiments on three systems. For each system we respectively list the ambient dimension n, the dimension $n-k$ of the solution set, and the degree d of the set. We list the average number of Newton iterations along a path for intrinsic and local intrinsic coordinates, as well as user cpu timings.

5 Conclusions

We list at least three advantages of using local intrinsic coordinates for sampling: (1) only the offset point moves; (2) the sparse structure of the polynomials is kept; and (3) we can control the step size by evaluation. Applications to numerical algebraic geometry include (1) implicitization via interpolation; (2) monodromy breakup algorithm; and (3) diagonal homotopies. In particular, local intrinsic coordinates will add to the robustness of our parallel subsystem-by-subsystem solver [14].

References

[1] E.L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*, volume 45 of *Classics in Applied Mathematics*. SIAM, 2003.

[2] E. Anderson, Z. Bai, C. Bischof, J. Blackford, S. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK's users guide*, volume 9 of *Sofware, Environments, and Tools*. SIAM, 3rd edition, 1999.

[3] J. Backelin. Square multiples n give infinitely many cyclic n-roots. Reports, Matematiska Institutionen 8, Stockholms universitet, 1989.

[4] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini: Software for numerical algebraic geometry. Available at `http://www.nd.edu/~sommese/bertini/`.

[5] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Adaptive multiprecision path tracking. *SIAM J. Numer. Anal.*, 46(2):722–746, 2008.

[6] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Software for numerical algebraic geometry: a paradigm and progress towards its implementation. In M.E. Stillman, N. Takayama, and J. Verschelde, editors, *Software for Algebraic Geometry*, volume 148 of *The IMA Volumes in Mathematics and its Applications*, pages 1–14. Springer-Verlag, 2008.

[7] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Stepsize control for path tracking. In D.J. Bates, G. Besana, S. Di Rocco, and C.W. Wampler, editors, *Interactions of Classical and Numerical Algebraic Geometry*, volume 496 of *Contemporary Mathematics*, pages 21–31. AMS, 2009.

[8] C. Beltran and Leykin. A. Certified numerical homotopy tracking. Preprint `arXiv:0912.0920v1 [math.NA]`.

[9] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

[10] B.H. Dayton and Z. Zeng. Computing the multiplicity structure in solving polynomial systems. In M. Kauers, editor, *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation (ISSAC'05), July 24-27 2005, Beijing, China.*, pages 116–123. ACM, 2005.

[11] J.W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

[12] P. Diaconis, D. Eisenbud, and B. Sturmfels. Lattice walks and primary decomposition. In B.E. Sagan and R.P. Stanley, editors, *Mathematical Essays in Honor of Gian-Carlo Rota*, volume 161 of *Progress in Mathematics*, pages 173–193. Birkhäuser, 1998.

[13] M. Griffis and J. Duffy. Method and apparatus for controlling geometrically simple parallel mechanisms with distinctive connections. US Patent 5,179,525, 1993.

[14] Y. Guan and J. Verschelde. Parallel implementation of a subsystem-by-subsystem solver. In *The proceedings of the 22th High Performance Computing Symposium, Quebec City, 9-11 June 2008*, pages 117–123. IEEE Computer Society, 2008.

[15] Y. Guan and J. Verschelde. PHClab: A MATLAB/Octave interface to PHCpack. In M.E. Stillman, N. Takayama, and J. Verschelde, editors, *Software for Algebraic Geometry*, volume 148 of *The IMA Volumes in Mathematics and its Applications*, pages 15–32. Springer-Verlag, 2008.

[16] S. Hosten and J. Shapiro. Primary decomposition of lattice basis ideals. *Journal of Symbolic Computation*, 29(4 and 5):625–639, 2000.

[17] M.L. Husty and A. Karger. Self-motions of Griffis-Duffy type parallel manipulators. In *Proc. 2000 IEEE Int. Conf. Robotics and Automation*, 2000. San Francisco, CA, April 24–28, CDROM.

[18] S. Kim and M. Kojima. Numerical stability of path tracing in polyhedral homotopy continuation methods. *Computing*, 73(4):329–348, 2004.

[19] A. Leykin. Numerical algebraic geometry for Macaulay 2. `arXiv:0911.1783v1 [math.AG]`.

[20] A. Leykin and J. Verschelde. Interfacing with the numerical homotopy algorithms in PHCpack. In N. Takayama and A. Iglesias, editors, *Proceedings of ICMS 2006*, volume 4151 of *Lecture Notes in Computer Science*, pages 354–360. Springer-Verlag, 2006.

[21] A. Leykin, J. Verschelde, and A. Zhao. Newton's method with deflation for isolated singularities of polynomial systems. *Theoret. Comput. Sci.*, 359(1-3):111–122, 2006.

[22] T.Y. Li. Numerical solution of polynomial systems by homotopy continuation methods. In F. Cucker, editor, *Handbook of Numerical Analysis. Volume XI. Special Volume: Foundations of Computational Mathematics*, pages 209–304. North-Holland, 2003.

[23] M.B. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter, J. McCarron, and P. DeMarco. *Maple Advanced Programming Guide*. Maplesoft, 2008.

[24] A. Morgan. *Solving polynomial systems using continuation for engineering and scientific problems*. Prentice-Hall, 1987. To appear in the SIAM Classics in Applied Mathematics Series.

[25] M. Petković. *Point Estimation of Root Finding Methods*, volume 1933 of *Lecture Notes in Mathematics*. Springer-Verlag, 2007.

[26] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical irreducible decomposition using PHCpack. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 109–130. Springer-Verlag, 2003.

[27] A.J. Sommese, J. Verschelde, and C.W. Wampler. Advances in polynomial continuation for solving problems in kinematics. *ASME Journal of Mechanical Design*, 126(2):262–268, 2004.

[28] A.J. Sommese, J. Verschelde, and C.W. Wampler. Homotopies for intersecting solution components of polynomial systems. *SIAM J. Numer. Anal.*, 42(4):552–1571, 2004.

[29] A.J. Sommese, J. Verschelde, and C.W. Wampler. An intrinsic homotopy for intersecting algebraic varieties. *J. Complexity*, 21(4):593–608, 2005. Festschrift for the 70th Birthday of Arnold Schönhage, edited by T. Lickteig and L.M. Pardo.

[30] A.J. Sommese, J. Verschelde, and C.W. Wampler. Introduction to numerical algebraic geometry. In *Solving Polynomial Equations. Foundations, Algorithms and Applications*, volume 14 of *Algorithms and Computation in Mathematics*, pages 301–337. Springer-Verlag, 2005.

[31] A.J. Sommese and C.W. Wampler. Numerical algebraic geometry. In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, pages 749–763. AMS, 1996. Proceedings of the AMS-SIAM Summer Seminar in Applied Mathematics. Park City, Utah, July 17-August 11, 1995, Park City, Utah.

[32] A.J. Sommese and C.W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific, 2005.

[33] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, 1999. Software available at `http://www.math.uic.edu/~jan/download.html`.